UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/686,882	10/15/2003	Kenneth Mark Williams	PA2594US	4103

22830 7590 02/27/2007

CARR & FERRELL LLP
2200 GENG ROAD
PALO ALTO, CA 94303

EXAMINER	
THOMAS, SHANE M	
ART UNIT	PAPER NUMBER
2186	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	02/27/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/686,882	WILLIAMS ET AL.	
	Examiner	Art Unit	
	Shane M. Thomas	2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 November 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *2-6,9-13,17-20 and 22-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *2-6,9-13,17-20 and 22-38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claims 24-38 are new while claims 1,7,8,14-16, and 21 have been canceled. As such, claims 2-6,9-13,17-20,22-38 are currently pending.

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.1 14, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.1 14, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114.

Excerpts from all prior art references cited in this Office action shall use the shorthand notation of [column # / lines A-B] to denote the location of a specific citation. For example, a citation present on column 2, lines 1-6, of a reference shall herein be denoted as "[2/1-6]."

All previously outstanding objections and rejections to the Applicant's disclosure and claims not contained in this Action have been respectfully withdrawn by the Examiner hereto.

Response to Amendments

Examiner has modified the prior art rejections in order to meet Applicant's amendments to the claims. Further new matter rejections under 35 U.S.C. 112, first paragraph, have been presented as discussed below in regards to a portion of the newly claimed subject matter.

Response to Arguments

Applicant has not submitted new arguments based on the amendments presented in the claims or in response to Examiner's previous rejections set forth in the prior final Office action filed 9/26/2006.

Claim Objections

Claims 18,19, and 36 are objected to because of the following informalities:

As per claims 18 and 19, the term --the number of data sequences-- should be amended to the term --the one or more data sequences--.

As per claim 36:

(1) the term --the system-- should be amended to --the computer system--;

(2) the term --data structures-- should be amended to --data sequences--; and

(3) the term --data sequence-- of line 5 should be amended to --data sequences--.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 24-33 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

As per claims 24 and 31, the Applicant does not teach the use of "executing a user defined instruction using the unaligned subset as an operand." Applicant recites the steps of using load and store instructions in combination with a register file to retrieve and store data in a register file (¶38); however Applicant's originally-filed specification is silent with regard to the use of unaligned data as an operand by a user-defined instruction. Further, Applicant's originally-filed specification teaches a configuration memory (¶43) for use with user-defined instructions, but does not elaborate and discuss using an unaligned subset of data as an operand for one of the user-defined instructions. In fact, Applicant's original disclosure is silent with regard to the term --operand-- altogether.

Claims 25-30, 32, and 33 are rejected as being dependent upon a rejected base claim.

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 2-6 and 34 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As per claim 34, it is not clear whether the term --the unaligned data sequence-- refers to the first of the --one or more unaligned data sequence-- or another of the --one or more unaligned data sequence-- other than the first data sequence, as the term --*the* unaligned data sequence-- lacks antecedent basis. Nonetheless, for the purposes of examination, the Examiner has considered the term --the unaligned data sequence-- to be the second unaligned data sequence read from the load/store buffer 24.

Claims 2-6 are rejected as they are dependent upon a rejected base claim.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 2,3,5,9,10,12,13,17-20,22,23, and 34-38 are rejected under 35 U.S.C. 103(a) as being unpatentable by Petersen (U.S. Patent No. 5,517,627) in view of Ramagopal et al. (U.S. Patent Application Publication No. 2003/0196058). Further the prior art references of Voith et al. (U.S. Patent No. 5,636,224) and Tanenbaum ("Modern Operating Systems") are being used to support an inherent feature of the Petersen and Ramagopal et al. references, respectively.

In order to provide a coherent rejection, the Examiner will follow claim dependence following each independent claim when discussing the claim rejections. For example, the Examiner will first discuss the claim rejections for independent claim 17 and all dependent claims before discussing independent claim 34 and its respective dependent claims.

As per claim 17, Petersen teaches a **system** (figure 8) comprising a **load/store buffer** 24 **configured to store data** (refer to figures 4A-4C and 5A-5B where it is shown that data bytes are stored in provided registers) and a **processor** 10 **configured to execute GET** (read) **instructions** (write and read aligners can be implemented with a ROM, which is well known in the art to contain instructions for controlling hardware systems, - [13/7-11]) for **processing data sequences.**

Alternatively, the Examiner could consider the memory controller 26 as being a

--processor-- as the memory controller can be seen in regard to figure 8 of Petersen to interact with the buffer memory 27 and the load/store buffer 24. Since the instructions to load/store the buffer 24 may be implemented via a ROM [13/8-11], it is necessarily inherent that some element perform the operation of executing the instructions stored on the ROM in order to utilize the load/store buffer 24. That element is being considered by the Examiner to be the --processor--.

Petersen teaches **initializing a load/store buffer** (data aligner 24, figure 8) **by loading a first aligned word** (4 bytes - [7/55-63]) **into the load/store buffer** (figure 5A - [8/41-47]) **from memory** 27 - [12/18-20]. The Examiner is considering loading a data word from the buffer 27 to the data aligner 24 as --initializing-- since the process places data into the data aligner. It can be seen with respect to figures 5A and 5B that it takes two loads to the data aligner 24 from the buffer memory 27 to --initialize-- the data aligner by placing data therein.

Further, Petersen teaches **further initializing the load store buffer** 24 **by loading a second aligned word** (figure 5B) **into the load/store buffer** [8/48-50] **from the memory**.

While Peterson **teaches reading one or more data sequences** (wherein a single data sequence is being defined by the Examiner as the number of bits transferred per lane L2(i), in the present embodiment of Petersen - one byte (8 bits) - [7/55-63]) **from the load/store buffer** 24, the reference does not specifically teach reading those data sequences **into a register file for instruction execution**. Peterson teaches that a DMA controller 13 is used to issue read and write requests on the host bus 14 to the peripheral device 15, thereby alleviating host processor from performing such duties [9/49 - 10/2], in addition to sending data from the system memory 12 to the peripheral device 15 [9/62-66]. The data retrieved or written to the buffer 27 is sent to the host bus 14 - [12/18-20].

Register files are well known in the processing and memory arts to be the collection of local processor registers **configured for instruction execution** (see noted prior art of record but not relied upon in the Conclusion section of this action).

Ramagopal teaches in ¶¶30-31 that data is received and sent between the memory 12 and register file 28 for instruction execution and storage after execution instruction, respectively (figure 1). Ramagopal further teaches that DMA accesses may be used to access memory 12 to store data and the like between the memory 12 and an external source (¶35). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the load/store buffering unit of Petersen with the teaching of associating a DMA access (DMA to/from external source to/from a memory that interfaces the register file) with a register file of Ramagopal in order to have (1) enabled the processor 10 of Petersen to have buffered the data to be executed in a memory (register file) very close in relation to the processor (to avoid having to continually fetch the data for each execution from the memory 12 during every instruction execution, thereby increasing instruction throughput) and (2) utilized the DMA processing to quickly send and receive data between the memory 12 and an external source (for example the peripheral device 15 of Petersen) for forwarding on to the register file - ¶35 of Ramagopal.

As a result of the combination of Petersen and Ramagopal, it can be seen that the **data sequences** from the **load/store buffer** 24 of modified Petersen would have been **read into the register file** (through the DMA controller's interaction between the memory 12 via the host bus 14 and the external device 15, comprising the load/store buffer 24), where the data sequences would have been **configured for instruction execution** - ¶35 of Ramagopal.

Further, as is known in the art of DMA accessing, the **reading of the one or more sequences from the load/store buffer into a register file** may be performed using an instruction receiving from the processor. The Examiner is considering this --instruction-- to be the number of bytes to transfer and the memory addresses involved that is commonly sent from the processor to the DMA device in order to initiate the DMA. Tanenbaum teaches such well-known DMA operation in lines 1-5 of page 31.

Peterson further teaches **at least one of the one or more data sequences being unaligned relative to the memory** [7/52-54].

Finally, Peterson teaches **loading additional aligned words to the load/store buffer from the memory** 27 **to replace the one or more data sequences that are read from the load/store buffer into the register file** ([13/51/54], where the reading process can be repeated as necessary to read data from the buffer memory 27 - [figure 13, step 153 - step 142]). The data sequences can be read into the register file as discussed above with reference to the Ramagopal teachings.

As per claim 18, Petersen teaches that **the number of data sequences read is an immediate specified number** that is specified by the host [8/62-67].

As per claim 19, Petersen teaches that **the number of data sequences read is a specified number stored as an index in a register memory** [9/4-6]. The Examiner is considering the value of the "CURRENT READ" to be an index.

As per claim 20, Petersen teaches **in which a first one of the one or most data sequences read is located at a first memory location** (i.e. any of the bytes of the 4 byte word entry of FIFO buffer 27 that is first read into the data aligner upon a read request from the host -

[8/41-46] that initializes the data aligner as discussed supra) **and the one or more data sequences comprises a specified number of data sequences stored in a register memory** 27 (here the Examiner is considering the buffer memory 27 as being the register memory and the --specified number-- of data sequences stored at the first index of the [FIFO] buffer 27 is specified to be four - as each [FIFO] buffer 27 entry contains four data sequences (four aligned bytes) - [1/26-35] and [8/30-50] - where the --first index-- is the index of the first entry of the FIFO buffer 27 to be read), **wherein the subsequent data sequence following the first of the data sequences is located at a second memory location pointed to by a second index** (the next memory location that is accessed from the FIFO buffer 27 - [8/48-50] as taught in the following example).

Petersen teaches that the first data sequence read can be any of the data sequences of the first FIFO buffer, with data unit 01 being used as an example that is read first [8/41-47]. If for example, the 04 data sequence was read first (i.e. the --first of the one or more data sequences--), thereby leaving data sequences 01,02, and 03, in the queues of the data aligner, data sequences 01, 02, 03, 05, 06, 07, and 08, would have been available to the host [8/51-52]. Thus it can be seen that the **subsequent data sequence** (data sequence 05) **following the first of the data sequences** (04) **is located at a second memory location** (buffer 27, second entry) that is **pointed to by a second index** (i.e. whatever the index/address of the FIFO buffer 27 contains the next data word that is read after the first data word - [8/48-50]).

As per claim 37, Peterson teaches **wherein the length of at least one of the one or more data sequences read from the load/store unit into the register file is less than a length of the first aligned word** (figures 5A-5B and [8/41-47]). The cited passages teach an example of

reading a data length of one byte (e.g. data unit 01) from the load/store buffer 24, which is less than the length of the first aligned work (e.g. four bytes) - [7/60-63].

As per claim 34, the rejection as well as the motivation to combine the Peterson and Ramagopal references follows the rejection and motivation set forth *supra* with regard to claim 17. Further, Peterson teaches **the alignment of the first and second word are relative to a memory** 27 **accessible to the processor** 10 - which is 4 data units [7/60-63], [8/42-47], [12/18-20], and [14/5-8]. Peterson additionally teaches **the unaligned data sequence including at least a part of the second aligned word** as taught in [8/52-55] with regard to the example posed in figures 5A-5B. Here, byte **05**, which is part of the second aligned word as shown in figure 5B, may be read if the processor requests a full width read (4 data units).

As per claim 2, Petersen teaches that **the data sequence length** [read] **is a byte** [8/42-43] and [7/60-61].

As per claim 3, Petersen teaches that **the data sequence length** [read] **is a bit** [7/64-67].

As per claim 5, Peterson teaches **wherein each of the one of the one or more unaligned data sequences has a length less than a length of the second aligned word** (figures 5A-5B and [8/41-47]). The cited passages teach an example of reading a data length of one byte (e.g. data unit 01) from the load/store buffer 24, which is less than the length of the first aligned work (e.g. four bytes) - [7/60-63]. It can be seen with reference to figures 5A-5B and [8/30-50] that the length of the second aligned word is the same length (e.g. 4 data units) as the first aligned word.

As per claim 35, the rejection follows similarly to the rejection of claim 34 above, except a write of data is occurring to the load/store buffer to be written again to the memory 27. The motivation to combine the references of Peterson and Ramagopal can be found above with respect to the rejection of claim 17.

Peterson teaches **a method for processing data sequences in a computer system comprising generating one or more unaligned data sequences** (column 5, Table 1, and [4/43-46]) **using an instruction** (e.g. a write instruction that sends data to be written to peripheral device 15 from the processor 10) **received from a processor** 10; **initializing a load/store buffer by loading data to the load/store buffer** (figures 4A-4C, Table 1, and [4/43-46]); **writing the one or more unaligned data sequences to the initialized load/store buffer** (figures 4A-4C, table 1, and [4/43-46], where the Examiner is considering --loading-- and --writing-- to be similar steps as data is written/loaded into the load/store unit); and **writing the one or more unaligned data sequences from the initialized load/store buffer to a memory** 27 [4/43-46]. The writing **uses a load/store instruction from the processor** (as discussed above, the processor may be used to execute the ROM instructions necessary to control the data aligner 24 [13/7-11] or the processor may use a DMA instruction to write data to the data aligner as previously discussed with respect to page 31 of Tanenbaum), **such that the one or more unaligned data sequences becomes aligned with the memory** 27 - [4/43-46] and [4/65-67].

As per claim 9, Petersen teaches that **the data sequence length** [written] **is a byte** - figure 4A, [6/53-59] and [4/7-10].

As per claim 10, Petersen teaches that **the data sequence length** [written] **is a bit** [4/9-10].

As per claim 12, Petersen teaches **flushing of the load/store buffer** 24 **in order to store any of the remaining unaligned data in order to store any of the remaining unaligned data into memory** (figure 4C and [6/64-7/4]). Figure 4C and [6/64-7/4] teaches that the unaligned data 01,02,03, and 04, now configured to an aligned word, is flushed (i.e. sent to the buffer 27 as taught above) while remaining unaligned data 05 is stored into memory (i.e. the queue(s) of the data aligner - [7/1-4]).

As per claim 13, Peterson teaches **wherein the length of each of the one or more data sequence is less than a length of an aligned word in the memory** 27 (figures 5A-5B and [8/41-50]). The cited passages teach an example of reading a data length of one byte (e.g. data unit 01) from the load/store buffer 24, which is less than the length of the first aligned work (e.g. four bytes) - [7/60-63].

As per claim 36, Peterson teaches **a system for processing data sequences in a computing system, where the computer system comprises a means for initializing a load store buffer** 24 **by loading one or more unaligned data sequences into the load/store buffer** (figures 4A-4C, Table 1, and [4/43-46]), **a length of each of the one or more unaligned data sequences being less than a length of an aligned word of a memory** 27 (figures 5A-5B and [8/41-50]). The cited passages teach an example of reading a data length of one byte (e.g. data unit 01) from the load/store buffer 24, which is less than the length of the first aligned work (e.g. four bytes) - [7/60-63]. Peterson teaches the computer system further comprises **the one or more unaligned data structures being unaligned relative to the memory** 27 [4/43-46], **and a means for writing the one or more unaligned data sequences to the memory** 27, **such that**

the written unaligned data becomes aligned to one or more word boundary of the memory 27 - [4/43-46] and [4/65-67].

Peterson does not specifically teach that the unaligned data sequence **includes an output of a user defined instruction executed using data retrieved from the memory**; however, Peterson does teach that a processor 10 may read and write data to a memory 27 via data aligner 24 using unaligned writes and reads [1/60-63]. It is well known in the art that processors access data from memory, manipulate data, and write data back to memory as needed based on the instruction flow of program being run by the processor. As such, it would have been obvious to one having ordinary in the art to have seen that that data that is written to the data aligner 24 from the processor 10 (figures 4A-4C and [6/53-63]) may be data that is the result (e.g. **output**) of a user defined instruction (e.g. any instruction of the program that the user of the computer system of modified Peterson is running with use of processor 10) executing using data retrieved from the memory 27, as the processor may access unaligned data from the memory 27 [7/46-54].

As per claim 38, Petersen teaches a **system** (figure 8) comprising a **load/store buffer** 24 **configured to store data** (refer to figures 4A-4C and 5A-5B where it is shown that data bytes are stored in provided registers) and a **processor** 10 **configured to execute PUT** (write) **instructions** (write and read aligners can be implemented with a ROM, which is well known in the art to contain instructions for controlling hardware systems, - [13/7-11]) for **processing data sequences in a computer system.** Peterson teaches **initializing the load/store buffer** 24 **by filling the load/store buffer with one or more unaligned data sequences** (figures 4A-4C, Table 1, and [4/43-46]), **a length of each of the one or more unaligned data sequence being**

less than a length of an aligned word (figures 5A-5B and [8/41-50]), and **the one or more unaligned data sequences being generated using part of the aligned word** ([4/43-46] teaches that the unaligned data sequences are part of an aligned word and when the combination of more than one unaligned sequences result in 4 data items, a aligned word is written to the memory 27. Finally Peterson teaches **writing one or more unaligned data sequences from the initiated load/store buffer** 24, **such that the unaligned data sequences become aligned relative to the aligned word** - [4/43-46] and [4/65-67].

Modified Peterson teaches by means of the Ramagopal reference that **unaligned data sequences** that are to be used to initialize load/store unit 24 can be **from a register file**. Refer to the rejection of claim 17 for the teachings of a register file and ¶30,¶31, and ¶35 of Ramagopal.

As per claim 22, Petersen teaches **the number of unaligned data sequences written is an immediate specified number** that is specified by the host [5/41-46], in similar fashion to the number of requested data sequences read [8/62-67].

As per claim 23, Petersen teaches that **the number of unaligned data sequences written is a specified number stored at an index in a register memory** where the --register memory-- is the combination of the registers S1(0) through S1(N) [6/16-20] with each register indicating that an unaligned data sequence is stored in the respective queue. In other words the specified number is an index since the registers S1(0) to S1(N) need to be indexed to acquire the number of unaligned data sequences that are queued in the data aligner 24 from a previous write request; these indexed registers (collectively a --register memory--) combine to produce the CURRENT QUEUED value.

Claims 4 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Petersen (U.S. Patent No. 5,517,627) in view of Ramagopal et al. (U.S. Patent Application Publication No. 2003/0196058), as applied to claims 1-3,5,6,8-10,12,13, and 15-23 above, in further view of Emma (U.S. Patent No. 5,619,665).

As per claims 4 and 11, Petersen teaches a general-purpose processor (host processor 10 - figure 7) but does not specifically teach a **processor having an extensible instruction set**. Emma teaches in the abstract that extension of an instruction set allows for circumventing software compatibility issues when allowing legacy software to benefit from new architectural extensions without recompilation and reassembly. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the data aligner system of figure 7 of Petersen with the teaching of an extension instruction set of Emma in order to have gained expandability of the instruction set of the processor 10 of Petersen, thereby gaining flexibility and compatibility for future instruction extensions without having to recompile or reassemble the system of Petersen.

Allowable Subject Matter

Claim 6 would be allowable if rewritten to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, set forth in this Office action and to include all of the limitations of the base claim and any intervening claims.

The following is a statement of reasons for the indication of allowable subject matter:

As per claim 6, the prior art of record does not teach nor reasonably suggest, either alone or in combination, changing a memory address pointer by an amount that is less than the length

of the second aligned word in order to point to the next unaligned data sequence that is to be read.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

(Previously cited) Hennessy et al. teaches a register file as being part of a processor on page 345.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane M. Thomas whose telephone number is (571) 272-4188. The examiner can normally be reached M-F 8:30 - 5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt M. Kim can be reached at (571) 272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane M. Thomas

PIERRE BATAILLE
PRIMARY EXAMINER
2/20/07